LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-CONF-205756

PortVis: A Tool for Port-Based Detection of Security Events

J. McPherson, K. Ma, P. Krystosk, T. Bartoletti, M. Christensen

August 5, 2004

VizSec/DMSEC 2004
Fairfax, VA, United States
October 29, 2004 through October 29, 2004

PortVis: A Tool for Port-Based Detection of Security Events

Jonathan McPherson Kwan-Liu Ma*
University of California at Davis

Paul Krystosk Tony Bartoletti Marvin Christensen †
Lawrence Livermore National Laboratory

Abstract

Most visualizations of security-related network data require large amounts of finely detailed, high-dimensional data. However, in some cases, the data available can only be coarsely detailed because of security concerns or other limitations. How can interesting security events still be discovered in data that lacks important details, such as IP addresses, network security alarms, and labels? In this paper, we discuss a system we have designed that takes very coarsely detailed data—basic, summarized information of the activity on each TCP port during each given hour—and uses visualization to help uncover interesting security events.

1 Introduction

Any network exposed to the Internet is likely to be regularly scanned and attacked by both automated and manual means. [11] Crackers will frequently scan entire ranges of ports, looking for open ports that can be exploited to gain access to a system. Worms and viruses often target specific ports in an attempt to locate systems that are vulnerable to the mechanisms they use to spread. These attacks are all recorded in security logs, but these logs are time-consuming for administrators to try to analyze by hand. Therefore, many attempts have been made to ease the detection of interesting information in the logs, using both traditional information visualization mechanisms like parallel coordinates, self-organizing maps, and multi-dimensional scaling, and novel visualization mechanisms designed specifically for this task. [5, 3]

Unfortunately, the level of attacks on a network is likely to be directly proportional to the value of the network. Networks that contain company or government secrets are more likely to be targeted by criminals inside or outside the network; large, high-profile networks make tempting targets for Internet terrorists. Therefore, network administrators can find themselves in a quandary when it comes to seeking outside network security help. They may not be permitted to reveal very much information about their networks' internal structure to security analysts, yet the analysts need a great deal of this information to do their jobs, since security visualization tools are likely to require very detailed data. For example, the *NAM* network security visualization system [4] requires information about *each individual packet* that goes across the network! Not all visualization systems require this level of detail, but most require, at the very least, IP address information.

Since information about the network's size, structure, and other important attributes may be sensitive, it is expedient to look at visualizations that permit network security events to be detected

*{mcpherso, ma}@cs.ucdavis.edu
†{krystosek1, azb, MarvinC}@llnl.gov

without the use of those attributes. This paper focuses on a visualization system that uses a very minimal set of attributes that reveals a minimal amount of information about the network. Both the system and the attacks and suspicious activity located with it will be discussed.

1.1 Related work

PortVis produces images of network traffic mainly by choosing axes that correspond to important features of the data (such as time and port number), creating a grid based on these axes, and then filling each cell of the grid with a color that represents the network activity there. This overall method of creating an image of network traffic is not wholly new; here is a (small) sampling of systems that function similarly to PortVis in this respect:

- *SeeNet* [1] uses an abstract representation of network destinations and displays a colored grid grid. Each point on the grid represents the level of traffic between the entity corresponding to the point's X value and the entity corresponding to the point's Y value.
- *NVisionIP* [7] uses network flow traffic and axes that correspond to IP addresses; each point on the grid represents the interaction between the corresponding network hosts. The points can represent changes in activity in addition to raw activity.
- [12] uses a quadtree coding IP of addresses to form a grid; Border Gateway Protocol (BGP) data is visualized as colored quadtree cells and connections between points on the quadtree.
- *The Spinning Cube of Potential Doom* [8] is a visualization system that uses two IP address axes and a port number axis to display network activity in a colorful, 3-dimensional cube. The combination makes attacks like port scans very clear; attacks that vary over the IP address space and port number produce interesting visuals (one method of attack, for instance, produces a "barber pole" figure).

1.2 Data

The data used in the examples in this paper comes from a collaborative working relationship with the Department of Energy. They have a number of network traffic analyzers installed at the Internet gateway of participating DOE sites. These traffic analyzers summarize large amounts of Internet Protocol (IP) traffic that flows to/from the Internet. For analysis purposes, this data is unclassified and is handled as Official Use Only (OUO). As a result of the summarization, the data is reduced to a set of counts of entities. For instance, instead of a list of each TCP session, there is a field that specifies how many TCP sessions are present; instead of a list of source IP addresses, a field specifies how many different source IP addresses were present.

The data is in the form of a large, space-delimited ASCII database table; the full list of fields present appears in Table 1. The first three fields are used for filtering and positioning the data; the last five fields are considered to be attribute values.

2 PortVis

2.1 Goals

PortVis was designed to achieve two goals:

Field	Example
Protocol	TCP
Port	80
Hour	2003-10-20 3:00am
Session count	1,443
Unique source addresses	342
Unique destination addresses	544
Unique source/destination address combinations	411
Unique source countries	20

Table 1: **The fields available to PortVis, and an example of each.** Each tuple represents *the activity on a given port during a given hour, through the given protocol.* The first three fields (Protocol, Port, and Hour) form a unique, composite key. The example row here is fictitious.

1. *Detect large-scale network security events.* PortVis should be able to permit analysts to discover the presence of any network security event that causes significant changes in the activity on ports. Since PortVis uses very high-level data, it is a very high-level tool, and is useful mostly for uncovering high-level security events. Security events that consist of small details—an intrusion that includes only a few connections, for instance—are unlikely to be caught using PortVis.

2. *Identify small-scale events for more detailed analysis.* Since PortVis only has counts of activities (rather than records of the activities themselves), its analysis can only go so far. It can identify suspicious traffic patterns, but it cannot see the traffic that caused the patterns. This is still useful, however; analysts using PortVis can send the suspicious traffic signatures to analysts that have access to the full set of network traffic logs. For instance, suppose that some port was completely unused during the first half of the time units available, but suddenly had a high level of activity for the remainder of the time units. Such an activity would seem to indicate the presence of a new entity on the network, and would likely be cause for further investigation.

2.2 Philosophy

PortVis was designed with a simple philosophy: *visualization generally flows from the highest-level semantic constructs to the lowest-level semantic constructs.* For instance, security experts might look at a *timeline* (high-level semantic construct) and discover that, during a particular hour, there was a lot of activity during a particular hour. They may then look at the specific *hour* (mid-level semantic construct) and discover that the activity was all concentrated on a particular port. They may then look at the specific *port* (low-level semantic construct) to examine the activity in the context of that port's normal activity, and discover that the activity is very anomalous, warranting an examination of the actual network traffic.

PortVis does not enforce this direction of flow rigidly, since it is possible that it may need to be reversed occasionally—for instance, if security experts make mistakes or see a lower-level pattern that they wish to see in the context of a higher-level visualization. To make it effortless to switch between the different semantic levels, they are all presented on the same screen. The user is not forced to pull down menus or deal with multiple windows or dialog boxes; all the controls and semantic levels are present at once—see Figure 1. Therefore, the cost of a context switch from

one semantic level to another is only that incurred by glancing at a new area on the screen and switching mental contexts; it does not include the time-consuming and possibly disruptive task of opening windows or dialogs.

There are three main semantic levels (already alluded to) used in PortVis: the *timeline*, the *hour* (main), and the *port*. Each has its own visualization.

2.3 The timeline visualization

The *timeline* is a visualization of the entire time range available to PortVis. The visualization has several elements, which are demonstrated in Figure 2:

The vertical axis (1) corresponds to *time*. Each row of the visualization represents one unit (generally an hour) of time. The top row is the earliest hour for which there is data; the bottom row is the latest hour for which there is data.

The horizontal axis (2) corresponds to *port range*. Each row consists of 32 blocks, each of which represents $65,536 \div 32 = 2,048$ of the ports. The leftmost block corresponds to the first 2,048 ports, the next block to the right corresponds to the next 2,048 ports, and so forth. The color of the block is determined by the level of activity on the ports during the time unit.

The selector (3) corresponds to *the currently selected time*. This is the time unit that is displayed on the main visualization panel.

The histogram (4) corresponds to *the relative frequencies of each activity level over the entire range of time*. "Activity level" here means "number of sessions." Therefore, if a very large number of ports have the same activity level, there will be a spike in the histogram at that activity level. The goal of the histogram is to provide information on activity levels so that they can be usefully mapped to colors. Note that all of the analyses of activity levels in the timeline window are done on a log scale; this is necessary because there are generally several ports with very high levels of activity (for instance, port 80), and these would irreparably skew a normal scale.

The gradient editor (5) corresponds to *the mapping from activity level to color*. The gradient editor can be used to explore spikes, gaps, or other interesting features of the activity level space revealed in the histogram by mapping each activity level to a smoothly interpolated color. Any number of arbitrarily colored control points can be added to the gradient; colors are linearly interpolated between control points. Figure 2 demonstrates the exploration of a histogram spike using the gradient editor. The spike turned out to correspond to a popular activity level for port scans. In general, operators are interested in seeing indications of port activity above certain levels [14], and the gradient editor can act as a filter to achieve this end.

This representation of the timeline works very well for analyzing up to several hundred hours of data at once, but as the number of hours approaches the number of rows of pixels available, detail is lost. Fortunately, alternative representations of time exist; for instance, [9] describes a method for compacting a timeline of arbitrary length into a visualization of constant size, and if PortVis is applied to larger data sets, it will need to acquire a similar capability.

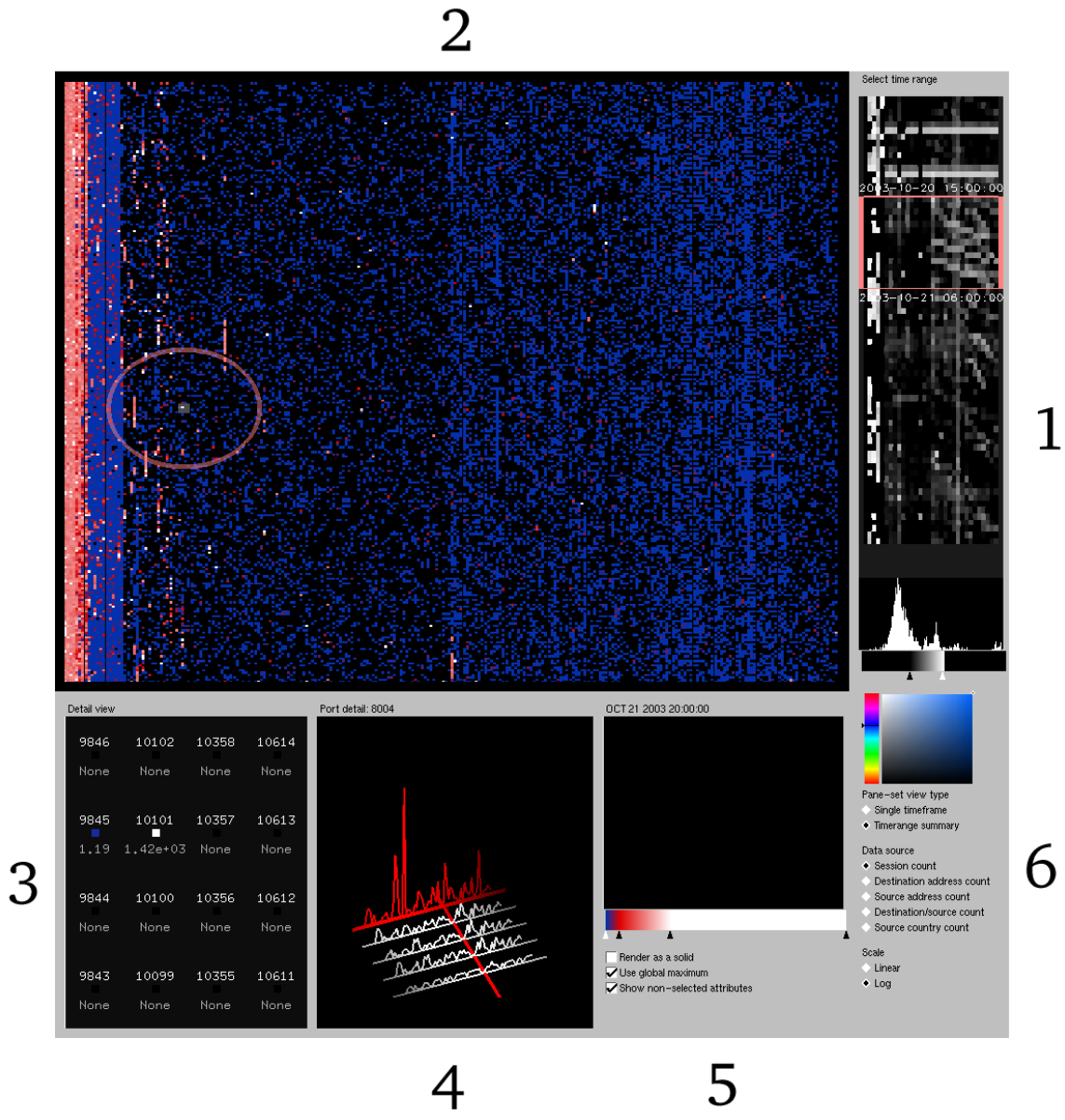

Figure 1: **The entire application.** Note that all of the available visualization tools are present simultaneously, so it is easy to correlate data and mentally shift between visualizations. Visualization generally begins at the *timeline* (1), followed by the hour (main) visualization (2). The main visualization contains a circle, which helps users locate the magnification square in its center. Magnifications from square within the main visualization are shown in (3); a port may be selected from (3) to get the port activity display in (4). Several parameters (5) control the appearance of the main display and port displays. The panel of options in (6) permits the selection of a data source to display, and offers a color-picker for selecting new colors for gradients.

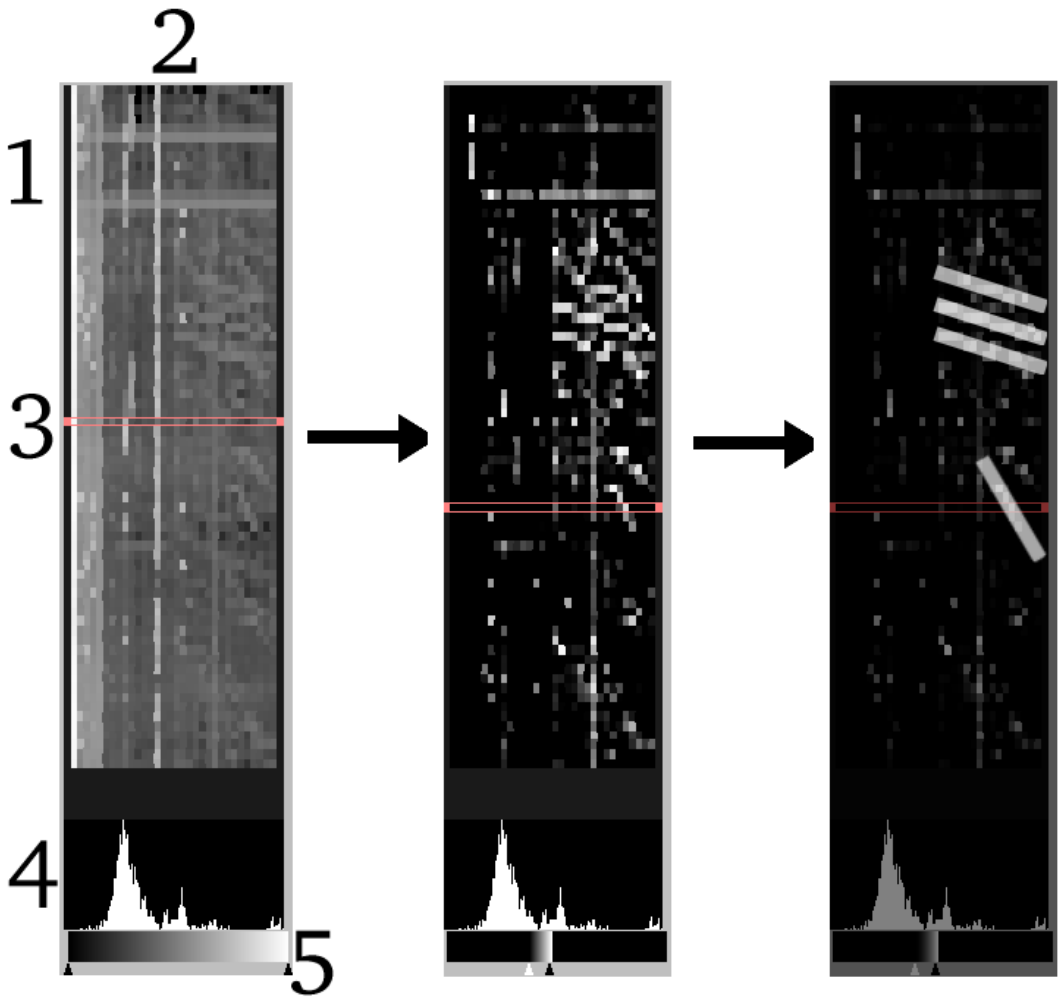

Figure 2: **The timeline visualization.** The vertical axis (1) represents time; the horizontal axis (2) represents port range. The selector (3) indicates the currently selected time. The histogram (4) represents the frequency of each level of port activity. The gradient editor (5) permits each level of activity to be mapped to a smoothly interpolated color. In this example, the gradient editor is used to emphasize a particular spike on the histogram, which leads to the discovery of possible port scans. The last image shows 4 possible port scans discovered; port scan awareness is very important to network security professionals. [14]

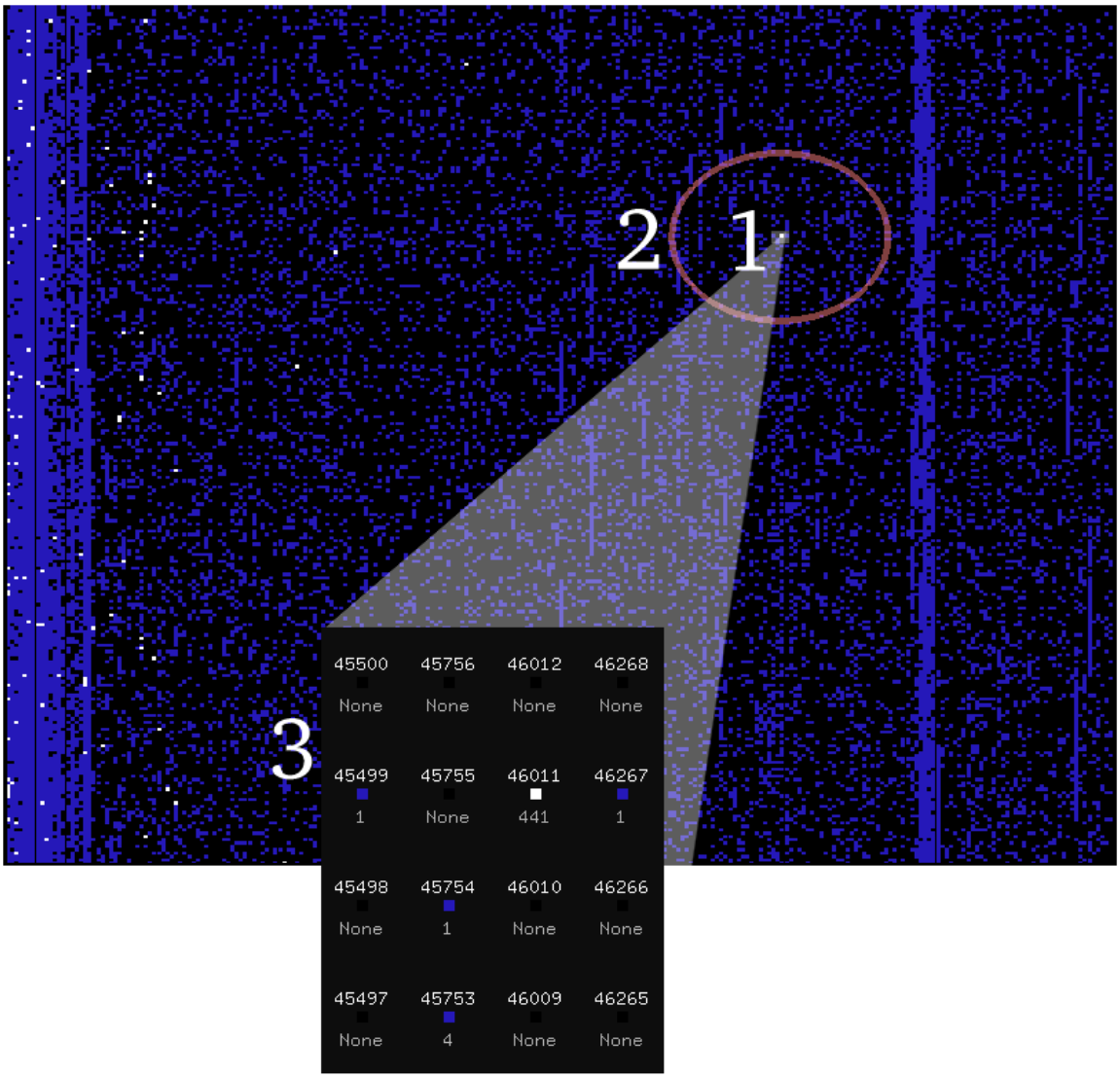

Figure 3: **The main visualization.** This data is from October 21, 2003. The high byte of the port number is the X axis, and the low byte of the port number is the Y axis. Each port's color represents the number of sessions on the port; dark blue ports have a low number of sessions, and white ports have a high number of sessions. The magnification tool allows users to select a small block of ports (1) to "magnify" in greater detail; a circle (2) is drawn around the magnified block to make it easier to find. The magnification is shown in (3); specific values are displayed for the selected attribute.

2.4 The main visualization

The *main visualization* depicts the activity during a given time unit. It consists of a dot on a 256×256 grid for each of the 65,536 ports. The dot's location on the grid (x, y) is calculated as follows, where *port* is the port number:

$$\begin{aligned} x &= port \div 256 \\ y &= port \bmod 256 \end{aligned}$$

The port number can be thought of as a two-byte number. Therefore, the X (horizontal) axis represents the *high byte* of the port number, and the Y (vertical) axis represents the *low byte* of the port number.

The following are the elements of the main visualization, which are demonstrated in Figure 3:

The vertical axis corresponds to the *high byte* of the port number (discussed above).

The horizontal axis corresponds to the *low byte* of the port number (discussed above).

Each point corresponds *a particular port*. The color of the point is determined by the numeric value at the port. A number of sources of data for the numeric values at ports can be selected; see Table 1 for a complete list. Points for which there exists no data (probably because there was no activity at all on the port) are always black.

A small, square selector (1) corresponds to *the ports currently being magnified*. The selector is 4×4 grid units in size and can be dragged around with the mouse to magnify any group of ports the user desires.

A large circle (2) serves to *help users locate the selector*. The selector is relatively small, and can easily get lost in the field of ports, especially when there is a lot of background noise.

A magnification area (3) serves to *provide detailed information about the magnified ports*. Each port's exact number is displayed, along with an enlarged visualization of its color point—to help users correlate it to the main visualization—and its exact data value.

A histogram (not shown) corresponds to the *the relative frequencies of each data value*. Like the histogram in the timeline, it serves to identify trends and/or patterns in the data.

A gradient editor (not shown) corresponds to the *the mapping from data values to colors*. Like the gradient editor in the timeline, it helps users explore gaps, spikes, and other interesting features that may be noticeable in the histogram.

2.5 The port visualization

The timeline visualization can identify a particular block of ports at particular time that warrant further investigation. The main visualization can often—as in Figure 3—identify the specific ports(s) to be investigated. But, given that information, one question remains: *is the identified activity on the port anomalous*? This question is addressed by the remaining visualization technique, which is a view of all the data available that concerns a particular port. The activity in question can be viewed in the context of all the activity on every attribute of the port, resolving

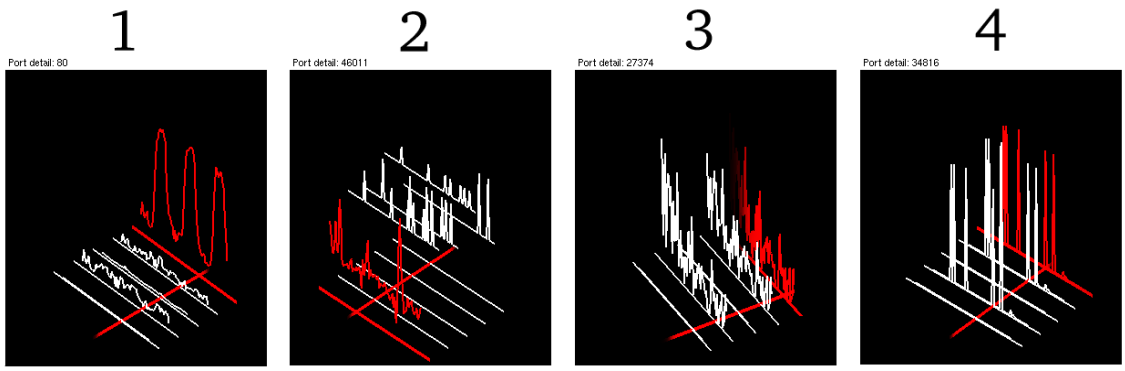

Figure 4: **The port visualization.** In each case, session count (the first attribute) is highlighted. These selected ports show a few distinct patterns of activity. The usage of Port 80 (1) is very periodic; it goes up during the day, and, predictably, down during the night. Port 46011 (2) has a fairly constant level of activity, with a few spikes. Port 27374 (3) is more erratic, though, interestingly, its usage drops noticeably as time goes on. Port 34816 (4) has one of the most suspicious usage graphs; it is only used a few times, but it is used fairly heavily during those times.

the question of whether the activity represents a deviation from the normal activity on the port. This port visualization is accessed by clicking on a port in the magnification area.

The port visualization has the following components, which are demonstrated in Figure 4:

The vertical axis corresponds to the *data values*; the greater the value, the more height.

The five straight lines each correspond to *one of the five attribute values* (Table 1 lists all the attributes). The attribute that is currently being analyzed with the main visualization is highlighted in red.

The remaining axis corresponds to *time*. The time currently being analyzed is indicated by a red bar.

The port visualization can be freely rotated about the vertical axis, which permits users to see it from the angle that reveals the details most interesting to them.

2.6 Comparing and contrasting

It is often the case that a network analyst is not interested so much in what occurred during a *particular* time unit but rather what *changed* across a *range* of time units. [7] Therefore, PortVis offers a feature that allows analysts to select any arbitrary set of time units and see on the main visualization not a depiction of the *actual* values at each port but rather a depiction of the *variance* of the values at each port. Suppose, for instance, that the analyst selected 4 hours, during which the port had 1,434 sessions, 1,935 sessions, 1,047 sessions, and 1,569 sessions, respectively. The system would then assign that port a value equal to the σ^2 of this set of values.

Figure 5 shows the variance analysis system in action. The security analyst has highlighted two regions of time that contain port scans in the upper regions of port space (see Figure 6 for a clear picture of the port scans).

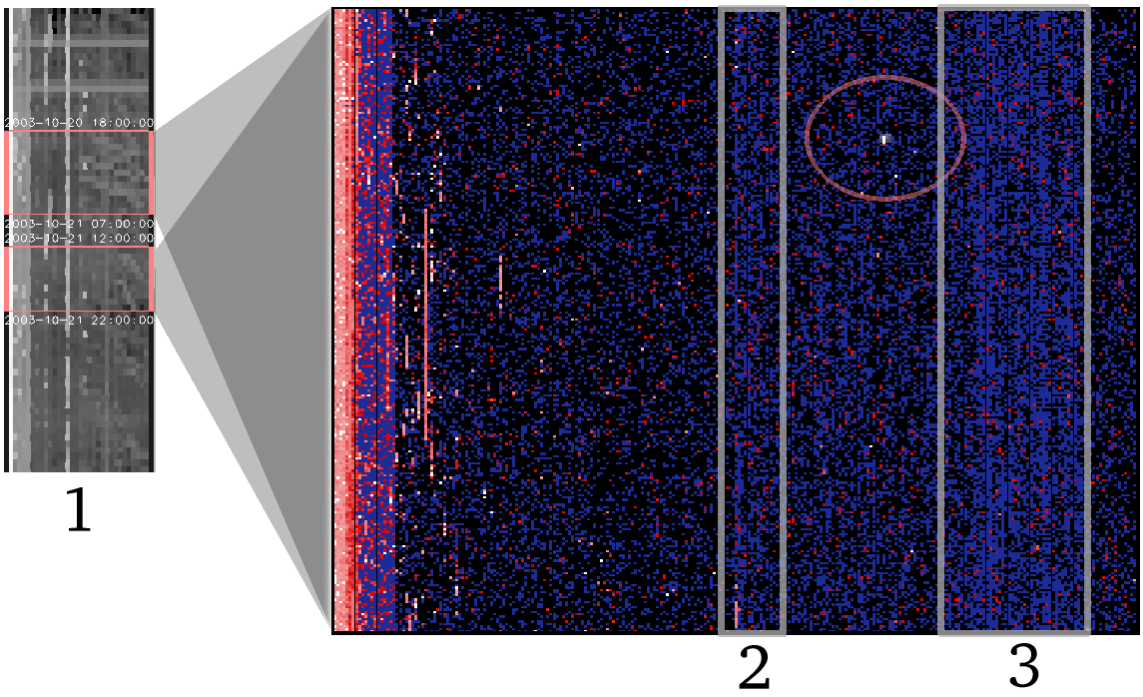

Figure 5: **The variance visualization.** Two ranges of time (1) have been selected. Black ports have no variance, meaning they had the same level of activity in all the time units selected. Blue ports have a very small level of variance. Red ports have a larger amount of variance, and white ports have the most variance. The most popular ports naturally have the most variance (see, for instance, the lower port range), but the really interesting feature of this image is the bands of ports with above-nominal variance—(2) and (3) are good examples. Clearly, something is causing large ranges of contiguous ports to all have about the same level of variance. Further analysis would reveal that this "something" is a scan of the ports.

3 Case studies

Some interesting patterns in the data have already been discussed; for instance, Figures 2 and 5 both depict the distinctive visual signatures caused by port scans; more detail is shown in Figure 6. PortVis, however, is able to visualize data that affects single ports as well as groups of ports. Figures 7 and 8 show how PortVis identifies ports for further analysis.

4 Conclusion

Even in settings where only generalized information is available concerning network activity, many types of malicious activity can still be discovered using visualization. We have developed a tool that takes general, summarized network data and presents multiple, meaningful perspectives of the data, and have demonstrated that this visualization leads to useful insights concerning network activity. Port scans of several types have been successfully detected, and many suspicious traffic patterns on individual ports have been uncovered. In addition, useful information about overall network traffic has been revealed; for instance, the rhythm of the traffic on commonly used ports as time progresses, and the relationships between the various metrics used to describe port activity.

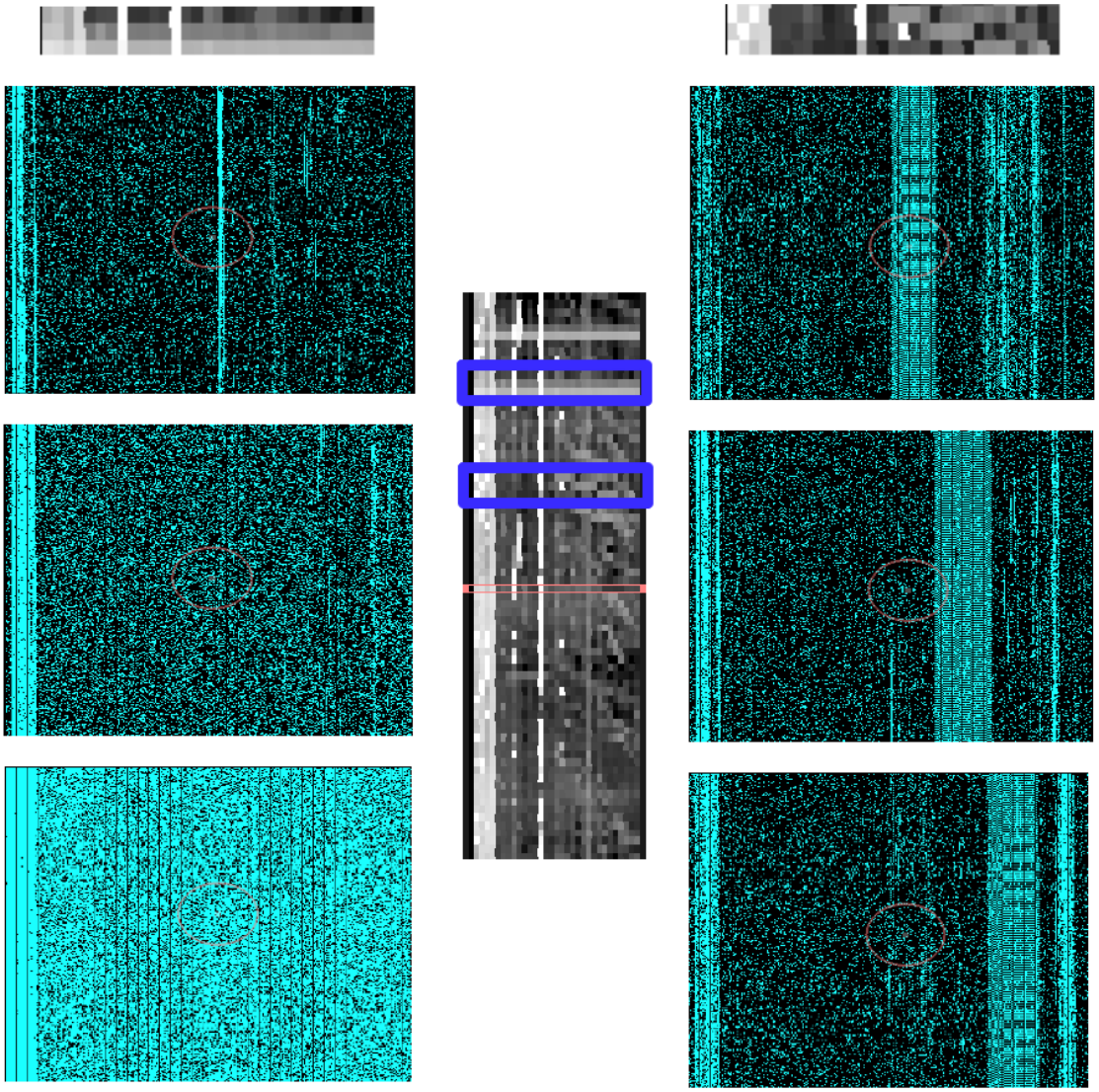

Figure 6: **Two port scans.** Two port scans are shown. Both started on October 20, 2003. The scan on the *left* is a linear scan, and ran from 11:00pm on October 20 to 2:00am on October 21. The scanning formed every-other-port stripes that covered most of the upper port range (the missed ports were covered in a subsequent scan, which is not shown here). The scan on the *right* is a "randomized" scan; from 10:00am–1:00pm on October 20, the scanner hit ports at random, eventually trying all of them. Network activity was fairly normal at 10:00am, but random port hits increased from 11:00am to 12:00pm, and between 12:00pm and 1pm, nearly every port had been hit. Note that both the randomized (top) and linear (bottom) scans stand out on the timeline, making them easy to tag for this kind of detailed analysis.

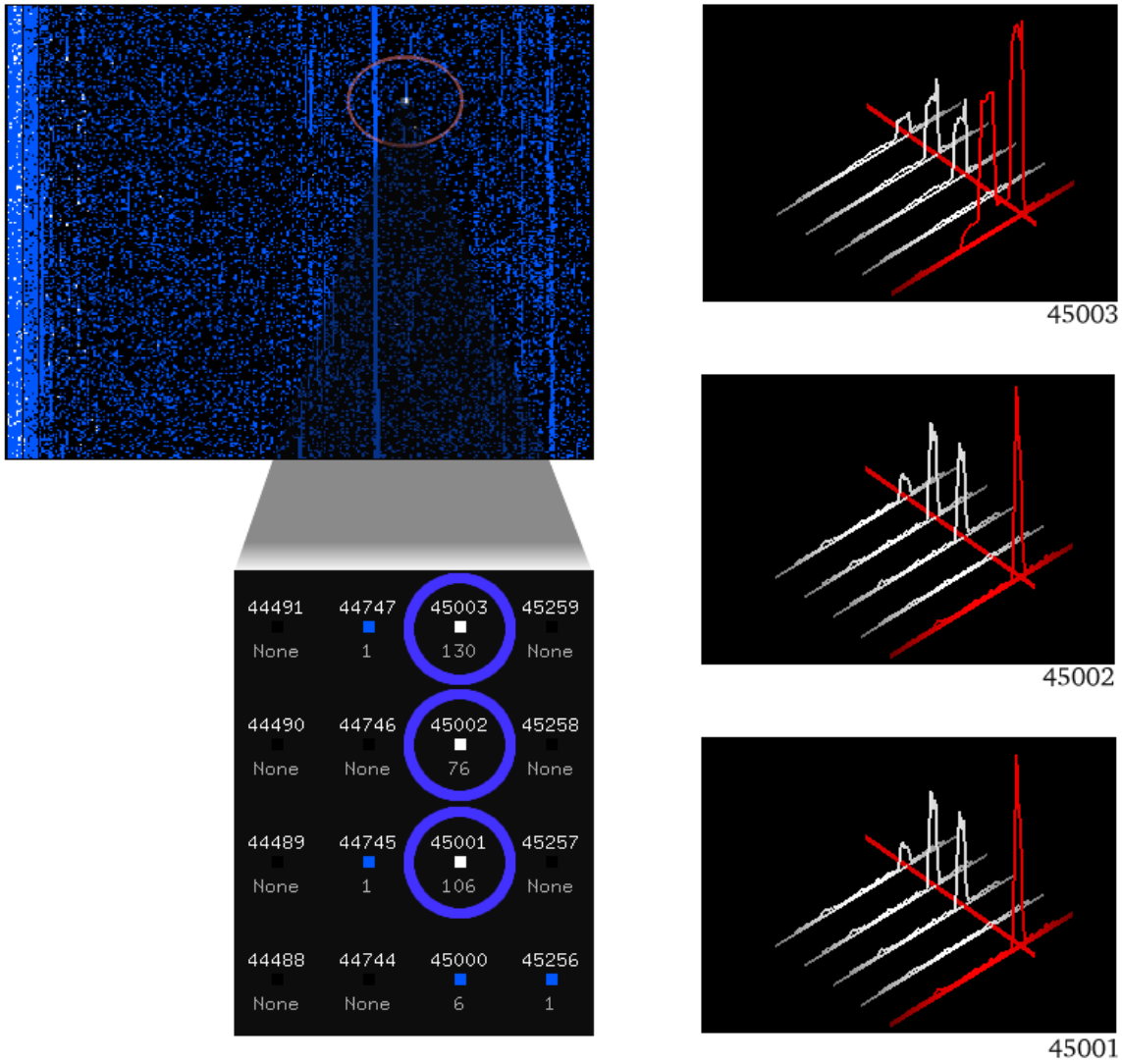

Figure 7: **Activity on three ports.** Between 5:00pm and 6:00pm on October 20, 2003, ports 45001, 45002, and 45003 had an anomalously high level of activity, causing them to appear in highlighted white on the main visualization. Three highlighted, sequential ports in a relatively unused segment of port space stood out enough to warrant analysis. All the activity on each port was displayed, showing that each port was relatively unused except for the burst of activity under investigation. It is impossible to ascertain what the actual traffic was using PortVis, but the pattern is suspicious.

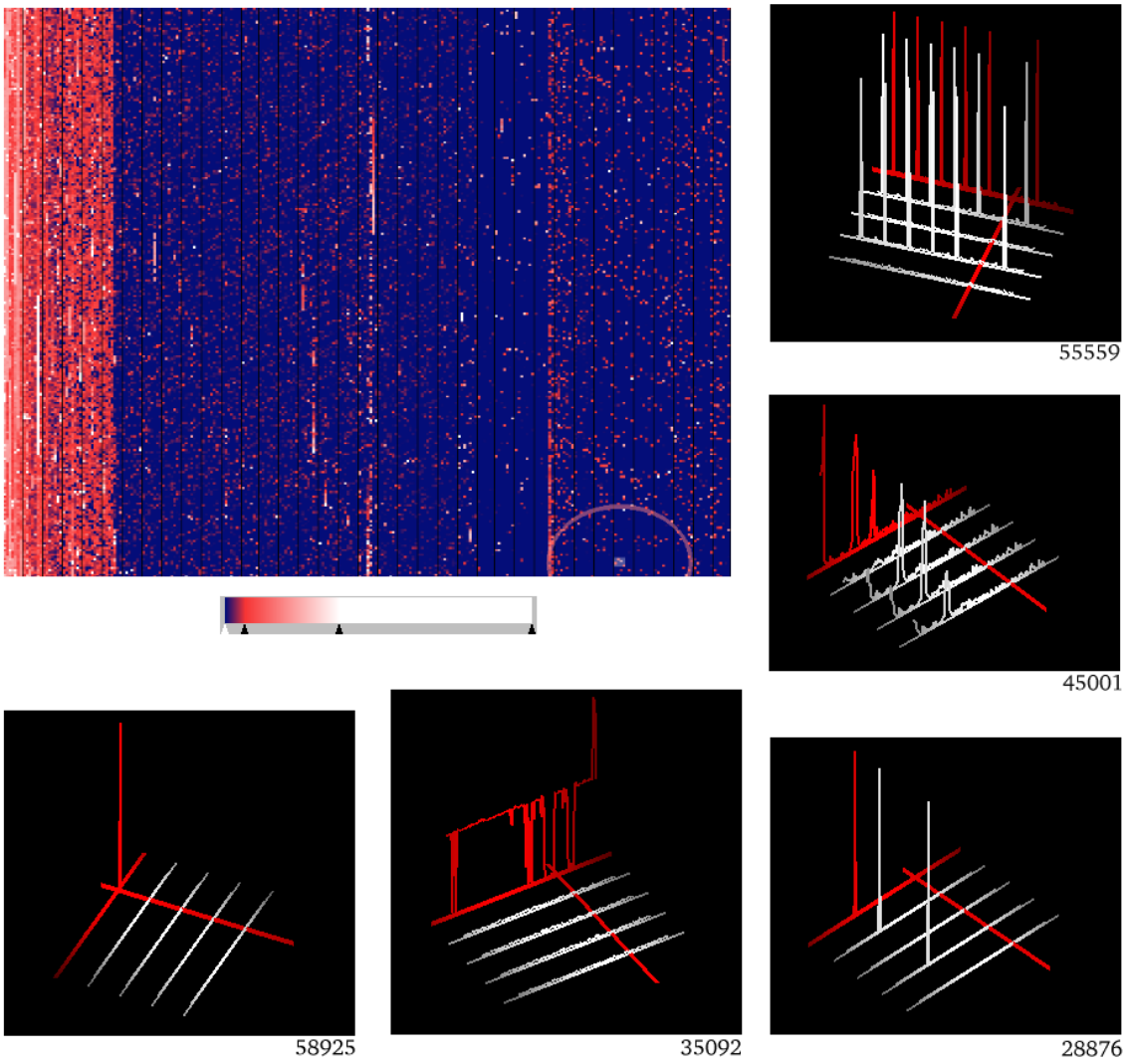

Figure 8: **Using variance to locate interesting ports.** This data is from a dataset that spans several days in February, 2004. The large image depicts the variance on each TCP port; it was generated by selecting nearly the entire time range and performing variance analysis on it. The variance was used to identify these 5 ports for further analysis. Ports 58925 and 28876 are typical of high-variance ports in this data; they have a single spike with a very high value, but no other interesting features. However, note that the session count spike on port 58925 does *not* have a corresponding spike in the destination counts, like the spike on port 28876 does. This indicates that the spike on 58925 is likely comprised of a suspiciously high level of network traffic between a small number of systems. Port 45001 has some spikes in session count that correspond to a rise in destinations, but some spikes that do not—again, a suspicious pattern. Other interesting patterns here are the "missed beat" on port 55559, and the "stutters" on port 35092.

5 Future work

5.1 Better use of existing attributes and integration of new attributes

The space of what can be accomplished with the raw attributes available has not yet been fully explored. Most of the visualizations presented in this paper focus on the raw level of activity on the port (session counts). However, interesting features may also lie in the other attributes and their correlations with each other. For instance, Figure 8 highlights the fact that there are ports with suspicious ratios of activity to destinations. This ratio, and others like it, could be used as quantities for visualization and analysis.

However, there is a limit to what can be done with summarized data; more interesting work lies in the integration of more detailed data about network activity. If IP addresses and other information about each session was available, the existing visualizations could be made much more richly detailed, and new visualizations could be created that could lead to insights that cannot be found in summarized data.

5.2 Machine learning

Currently, human pattern detection is relied upon to find patterns in the data and groups of related ports. However, machine learning could be potentially applied to find patterns and anomalies, augmenting human abilities. Since PortVis focuses on unlabeled data, clustering algorithms are likely to be of use, since these have proven to be useful in discovering security events in unlabeled data. [10] For instance, a self-organizing map [6] or multi-dimensional scaling [13] could be used to organize the ports according to their nearness in data space (similar to [5]), hopefully isolating the ports with unusual usage. Another machine learning approach to finding interesting outliers is discussed in [2].

5.3 User interface improvements

The user interface could be improved in a number of ways. Many of the controls require fine adjustment because the areas of interest can be as small as several pixels in size; this could be resolved by permitting zooming in the areas of the interface that have small features. Also, the system should have the capability to save and restore visualization states, so that interesting views could be easily recalled. Very useful views could evolve into a kind of "at-a-glance" network visualization system. The system's responsiveness could be improved; currently, it reads data from the raw text files and computes its statistics. It would save the user time if the data were pre-processed and stored so that data loaded more quickly upon startup. The method of display for detailed port information may do better as a group of 2-dimensional graphs rather than one 3-dimensional graph, since occlusion can be a problem. It would be useful to have the ability to look up information from the Web or an internal database about the ports commonly used by popular programs and services, to avoid false alarms generated by benign network entities.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

References

[1] Richard A. Becker, Stephen G. Eick, and Allan R. Wilks. Visualizing network data. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):16–28, 1995.

[2] P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Srivastava, and P. Tan. Data mining for network intrusion detection. In *Proc. NSF Workshop on Next Generation Data Mining*, 2002.

[3] Robert F. Erbacher. Visual traffic monitoring and evaluation. In *Proceedings of the Conference on Internet Performance and Control of Network Systems II*, pages 153–160, 2001.

[4] Deborah Estrin, Mark Handley, John Heidemann, Steven McCanne, Ya Xu, and Haobo Yu. Network visualization with NAM, the VINT network animator. *Computer*, 33(11), 2000.

[5] L. Girardin and D. Brodbeck. A visual approach for monitoring logs. In *Proceedings of the 12th Usenix System Administration conference*, pages 299–308, 1998.

[6] Teuvo Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin, 3rd edition, 1989.

[7] Kiran Lakkaraju, Ratna Bearavolu, and William Yurcik. NVisionIP—a traffic visualization tool for security analysis of large and complex networks. In *International Multiconference on Measurement, Modelling, and Evaluation of Computer-Communications Systems (Performance TOOLS)*, 2003.

[8] Stephen Lau. The spinning cube of potential doom. *Communications of the ACM*, 47(6):25–26, 2004.

[9] K. Mundiandy. Case study: Visualizing time related events for intrusion detection. In *Proceedings of the IEEE Symposium on Information Visualization 2001*, pages 22–23, 2001.

[10] Leonid Portnoy, Eleazar Eskin, and Salvatore J. Stolfo. Intrusion detection with unlabeled data using clustering. In *Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001)*, 2001.

[11] S. Staniford, V. Paxson, , and N. Weaver. How to own the internet in your spare time. In *Proceedings of the 2002 Usenix Security Symposium*, 2002.

[12] Soon Tee Teoh, Kwan-Liu Ma, S. Felix Wu, and Xiaoliang Zhao. Case study: Interactive visualization for internet security. In *Proc. IEEE Visualization*, 2002.

[13] F. W. Young and R. M. Hamer. *Multidimensional Scaling: History, Theory and Applications.* Erlbaum, New York, 1987.

[14] William Yurcik, James Barlow, Kiran Lakkaraju, and Mike Haberman. Two visual computer network security monitoring tools incorporating operator interface requirements. In *ACM CHI Workshop on Human-Computer Interaction and Security Systems (HCISEC)*, 2003.